

# Comparison of optimizer-based and flow solver-based trimming in the context of high-fidelity aerodynamic optimization

Časlav Ilić

**Abstract** This report compares two approaches for achieving a trimmed state of an aircraft configuration during an aerodynamic optimization. In the optimizer-based approach, balance equations are posed as direct constraints to the optimizer. In the flow solver-based approach, balance equations are satisfied within the flow solver evaluation. These approaches are applied to a flying wing case, where blended trailing edge deflection is used to control the pitching moment. The wing is treated as rigid, and lift and pitching moment balance equations are taken into account for trimming. Tests are performed with varying numbers of shape design parameters and with varying numbers of flight points. It is concluded that the flow solver-based approach performs more robustly, and thus should be preferred in general, even though it may take more time than the optimizer-based approach.

## 1 Introduction

When performing an aerodynamic optimization of an aircraft configuration in steady flight, it is usual to require some force balance conditions to hold. For typical transport aircraft in horizontal flight, these conditions are (neglecting small angles) that aircraft weight must be balanced by aerodynamic lift, aerodynamic pitching moment around the center of gravity by horizontal tail lift, and aerodynamic drag by engine thrust. When these forces are in balance, it is said that the aircraft is in a *trimmed* state.

Previous paragraph assumes that the aircraft is modeled as a rigid body; if structural elasticity were modeled too, internal structural and inertial forces would have to be balanced with external forces as well. Furthermore, in this report the drag-thrust balance is taken to be implicitly satisfied, because an active engine is not

---

Časlav Ilić, e-mail: caslav.ilic@dlr.de  
German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology,  
Lilienthalplatz 7, 38108 Braunschweig, Germany

modeled. This leaves only the weight–lift and pitching moment–tail balances. The report by Merle et al. [6], which focuses on adjoint-based aerodynamic optimization in presence of active engine boundary conditions, does include explicit drag-thrust balance, and compares on the test case used therein the trimming approaches examined here.

There are two approaches to achieve trimmed state during the optimization. The first, more straightforward, is to pose required force balances as constraints for the optimizer, to be satisfied by the end of optimization [2]. The other, more involved approach is to satisfy the force balances at each design evaluation by an internal iteration within the aerodynamic flow solver [1]. In this report, the former approach is called the *direct optimization strategy* (DOS) and the latter the *trim-corrected optimization strategy* (TCOS). In DOS, trim control parameters – angle of attack  $\alpha$  and tail deflection angle  $\delta$  – are included as design parameters in the optimization, together with the shape design parameters. With the goal to minimize drag (which implies minimizing engine thrust, and thus fuel consumption), this approach can be formulated as

$$\begin{aligned} \min_{\tilde{p}} \quad & C_D(u(\tilde{p})), \\ \text{subject to} \quad & C_L(u(\tilde{p})) = C_L^*, \quad C_M(u(\tilde{p})) = 0, \\ & \text{with flow solver satisfying } R(u(\tilde{p})) = 0, \end{aligned} \tag{1}$$

where  $\tilde{p} = (p, \alpha, \delta)$  is the composite vector of shape design parameters  $p$  and trim control parameters,  $C_D$  is the drag coefficient,  $C_L$  the lift coefficient,  $C_M$  the pitching moment coefficient,  $R$  the residual of flow equations and  $u$  the flow state variables. In TCOS,  $\alpha$  and  $\delta$  are treated as flow state parameters, i.e. the force balance equations are solved during the design evaluation, together with the flow residual equations. This approach is formulated as

$$\begin{aligned} \min_p \quad & C_D(\tilde{u}(p)), \\ \text{with flow solver satisfying} \quad & R(\tilde{u}(p)) = 0, \quad C_L(\tilde{u}(p)) = C_L^*, \quad C_M(\tilde{u}(p)) = 0, \end{aligned} \tag{2}$$

where  $\tilde{u} = (u, \alpha, \delta)$  is the composite vector of flow state variables and trim control parameters.

Why should one consider TCOS at all, given that it solves the same problem, but requires additional implementation effort compared to DOS? One reason may be historical, where existing trimming functionality in and around the flow solver is carried over into an optimization context. Another reason may be using an optimization algorithm which cannot handle constraints, as is frequently the case with derivative-free algorithms. In this work, however, a gradient-based algorithm which does handle constraints is used.

More fundamentally, TCOS exploits special knowledge about the problem. It identifies for each trimming constraint (force balance equation) one specific parameter of by far the highest influence, and handles those parameters in a subproblem, leaving the optimizer to deal with other, non-specific parameters. In case of a single

local optimum and the starting point (baseline design) inside the convex region of that optimum, this should not produce an advantage over DOS. In fact, here TCOS should even increase the run time of optimization, since it expends effort to satisfy trimming constraints exactly at each point, even while far away from the optimum. However, in cases of multiple local optima and non-convex paths from the starting point, TCOS should be more robust, since the trimming subproblem alone is well-behaved for usual aerodynamic shapes. Similarly, TCOS should be more robust in face of (intentionally or unintentionally) reduced accuracy of gradients. The question then is what is the extent of these tradeoffs between DOS and TCOS.

## 2 Methodology

For evaluating aerodynamic characteristics of a configuration under optimization, the DLR TAU [10] flow solver is used. Transonic flow is modeled with Reynolds-averaged Navier-Stokes (RANS) equations and Spalart-Allmaras 1-equation turbulence model.

Gradients of the aerodynamic cost functions (lift, drag, and pitching moment coefficients) are computed with the help of the discrete adjoint method implemented within TAU. Details about implementation and validation studies can be found in papers by Dwight [4] and Widhalm [12]. Here it is sufficient to state that with the adjoint method it is possible to compute the derivative of a cost function with respect to all design parameters in time practically independent of the number of design parameters, and using the same computing resources as for the flow computation. Rather, gradient computation time depends only on the number of cost functions. This is a major enabler for gradient-based aerodynamic optimization, as it typically requires few cost functions but large number of design parameters.

For parametrizing the shape of the configuration, the free-form deformation (FFD) [8] is used. It works by first enveloping the surface mesh points of the baseline configuration with a lattice of non-uniform rational basis spline (NURBS) control points, and projecting each surface point spatial coordinates to the its NURBS coordinates. These are kept fixed. Then, when a NURBS control point is moved during optimization, new spatial coordinates of points are recomputed from their NURBS coordinates. Figure 1 shows the FFD lattice around the Onera M6 wing which is used a test case in this report.

Another consideration is how to change the shape such as to achieve zero pitching moment around the center of gravity. For a conventional wing-body-tail configuration, this is typically done by applying a rotational deformation of the horizontal tail, which is dependent only on the tail deflection parameter  $\delta$ . In this report, however, since a standalone wing is used, pitch trimming is performed by deflecting the wing trailing edge, starting from 70% chord, along the complete span. Between the virtual hinge point and the trailing edge, deflection function is parabolic rather than linear, to avoid introducing a sharp edge on the surface mesh.

When the surface mesh deformation is computed, mesh points in the volume are moved using the linear elasticity approach [3]. This approach is known to produce high quality deformed meshes, at some time cost due to having to solve a usually stiff linear problem (caused by the high aspect ratio of cells needed in RANS-capable meshes).

In TCOS optimizations, trimming is performed in the following way. Angle of attack  $\alpha$  in the free-stream boundary condition is updated each fixed small number of flow solver iterations, using a quasi-secant method on the lift balance equation. Here the derivative of lift wrt.  $\alpha$  is not estimated through the iteration, rather a fixed value of 0.01 is used, because usual aerodynamic shapes are known to exhibit nearly linear dependence between lift and angle of attack. The situation with the pitching moment balance equation is somewhat different. The dependence between pitching moment and deflection angle  $\delta$  is also usually linear, however here the mesh must be deformed for each change in  $\delta$ , which is a costly operation. Thus,  $\delta$  is updated with a much longer iteration period than  $\alpha$ , to the point that the lift balancing appears like a subiteration within the pitching moment balancing. For pitching moment balancing an actual secant method is used, where the derivative of pitching moment wrt.  $\delta$  is approximated from the last two steps. For the used test case, it takes from 3 to 5  $\delta$  updates to reduce absolute value of the pitching moment coefficient below  $10^{-5}$ .

A crucial element of TCOS is correction of the objective function gradient wrt. shape design parameters as delivered by the adjoint method. In TCOS problem formulation (2)  $\alpha$  and  $\delta$  are functions of shape design parameters  $p$ , so the gradient of the drag coefficient can be written as

$$\frac{dC_D(p, \alpha(p), \delta(p))}{dp} = \frac{\partial C_D}{\partial p} + \frac{\partial C_D}{\partial \alpha} \frac{d\alpha}{dp} + \frac{\partial C_D}{\partial \delta} \frac{d\delta}{dp}. \quad (3)$$

Since lift the coefficient and the pitching moment are kept constant (the latter being zero) by the trimming procedure, it also holds that

$$\frac{dC_L(p, \alpha(p), \delta(p))}{dp} = \frac{\partial C_L}{\partial p} + \frac{\partial C_L}{\partial \alpha} \frac{d\alpha}{dp} + \frac{\partial C_L}{\partial \delta} \frac{d\delta}{dp} = 0, \quad (4)$$

$$\frac{dC_M(p, \alpha(p), \delta(p))}{dp} = \frac{\partial C_M}{\partial p} + \frac{\partial C_M}{\partial \alpha} \frac{d\alpha}{dp} + \frac{\partial C_M}{\partial \delta} \frac{d\delta}{dp} = 0. \quad (5)$$

Terms  $\partial C_*/\partial p$  are the derivatives computed by the adjoint method (and used directly in DOS). Terms  $\partial C_*/\partial \alpha$  are the derivatives on the boundary condition, and are derived inside TAU and computed as part of solving the flow adjoint equation. Since they involve mesh deformation, terms  $\partial C_*/\partial \delta$  are computable by the same adjoint method as terms  $\partial C_*/\partial p$ , however this was not yet implemented in the applied evaluation process. Instead, they are computed via central finite differences, after the trimmed solution has been found. (Another idea might be to use the last approximation of these terms in the secant method applied to the pitching moment balance, but these values are only first-order accurate.) With all these terms available, equations (4) and (5) can be solved as a system of equations to obtain terms

$d\alpha/dp$  and  $d\delta/dp$ , which are then substituted into equation (3) to obtain the final objective function gradient.

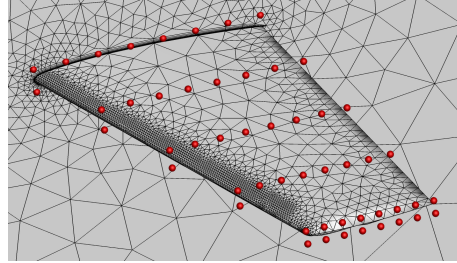
A sequential-quadratic programming (SQP) algorithm is used to drive the optimization, in form of the widely applied SNOPT [5] code. SNOPT is linked to the objective value and gradient evaluation routines using the pyOpt [7] optimization framework. The complete setup is run on an high-performance computing (HPC) cluster.

### 3 Results

The test case under examination is the Onera M6 wing [9], one of the standard transonic flow test cases in computational aerodynamics. It has a low aspect ratio of 3.8, leading edge sweep of  $30^\circ$ , and symmetric airfoils. In this report, the design flight point for optimization is set to Mach number 0.84, Reynolds number of  $12 \cdot 10^6$ , lift coefficient  $C_L = 0.20$ , and pitching moment coefficient  $C_M = 0$  at 25% mean aerodynamic chord. As mentioned earlier, the pitching moment is balanced by deflecting the trailing edge from 70% chord, along the full span, and using a parabolic blend function.

The shape is parametrized using an FFD lattice, as show on figure 1. Only z-coordinates of FFD control points are design parameters, so the planform is never changed. Furthermore, each pair of above- and below-wing control points are moved together by a single parameter, to avoid reduction of airfoil thicknesses. Finally, the control point pairs at leading edge and trailing edge of the wing root are fixed, to avoid implicit rotation or translation of the complete wing.

**Fig. 1** Onera M6 wing hybrid-unstructured mesh equipped with a  $5 \times 8 \times 2$  lattice of FFD control points.



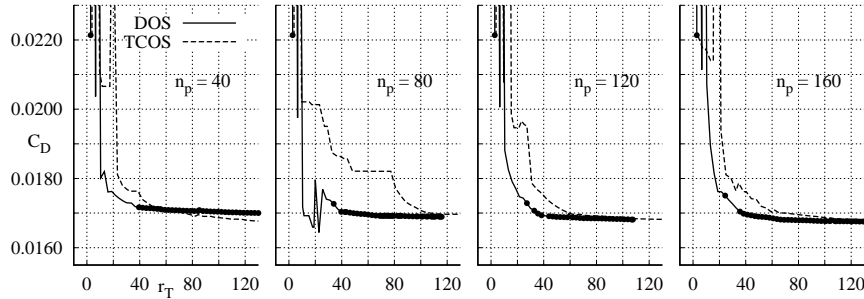
Before presenting the performed optimizations, first it should be explained how DOS and TCOS are going to be compared. In case the problem had a single local optimum, the only needed comparison would be the time needed to reach that optimum. But even here one must be careful, since timings can be very much implementation dependent. For example, in the present case, the process parameters in the trimming procedure in TCOS were a bit too conservative, increasing run time likely by 50% or more; then, the finite differencing for  $\partial C_*/\partial \delta$  terms took almost half of the complete gradient computation time. Instead we assume that, in a reason-

ably effective implementation, flow evaluation with trimming procedure in TCOS would need factor 2 time relative to pure flow evaluation in DOS, and that the adjoint gradient computation per cost function in both TCOS and DOS would take factor 0.6 relative time to flow evaluation (split as 0.5 for flow adjoint and 0.1 for mesh adjoint solving, while TCOS correction takes negligible time), for a complete gradient computation relative time of 1.8 for three cost functions. Thus, the x-axis of convergence curves will accumulate this estimated relative time factor  $r_t$  to single pure flow evaluation. DOS runs start from trimmed solutions, obtained by trimming of the baseline design. DOS convergence curves also have feasible designs marked, where a design is declared feasible if both  $|C_L - C_L^*|$  and  $|C_M|$  are below  $10^{-4}$ . In TCOS runs all designs are feasible by definition.

However, as will be seen, the used test case apparently does not have a single local optimum, and this is not unexpected in transonic flows. In this case, in principle, run times can depend on the optimum to which, by chance, particular optimization run happens to converge to. Still it will be possible to extract some conclusions.

Two series of optimizations are performed: one series of single-point optimizations with varying number of design parameters, and another series with varying number of flight points at fixed number of design parameters.

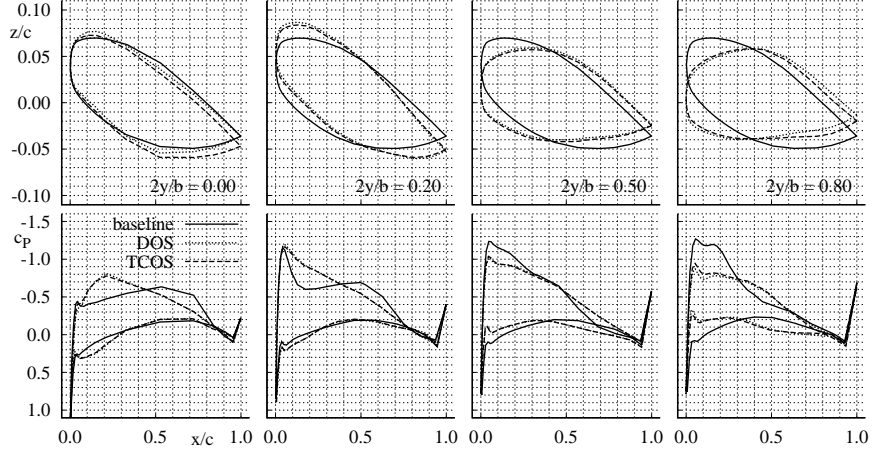
Number of design parameters  $n_p$  is varied as 40, 80, 120, and 160. The optimization convergence curves are shown on figure 2. It can be seen that for every  $n_p$  DOS reaches feasibility about when it also reaches convergence in the objective, and that TCOS takes about 50% more run time. Both DOS and TCOS reach same quality of optima. The exception is the  $n_p = 80$  case, where TCOS takes considerably more time, but manages to recover and reach the same objective value. This is likely the consequence of the highly non-convex, possibly multiple-optima nature of the transonic flow.



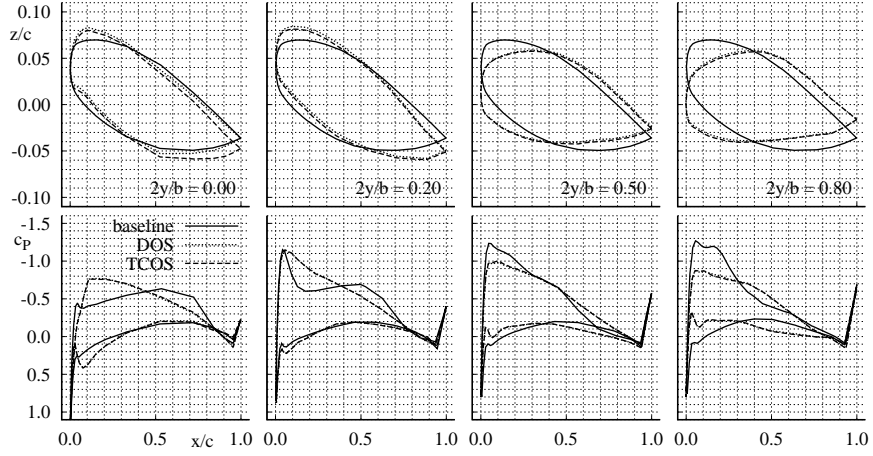
**Fig. 2** Comparison of single-point DOS and TCOS optimization convergence with varying number of design parameters  $n_p$ . Dots on DOS curve represent feasible designs.

Figures 3 and 4 compare (trimmed) baseline and optimized airfoil sections at four relative spanwise locations  $2y/b$ , with airfoils also rotated to optimized angle of attack. It can be seen that indeed the optimal shapes differ, both between DOS and TCOS (especially near the root and the tip), and between different number of design

parameters, even though their objective values as well as pressure distributions are quite near.



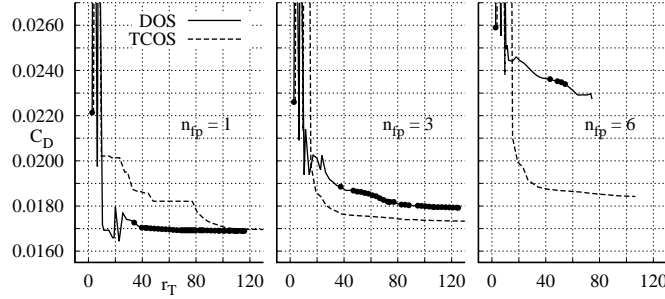
**Fig. 3** Comparison of airfoils and pressure distributions for single-point DOS and TCOS optima with  $n_p = 80$  design parameters



**Fig. 4** Comparison of airfoils and pressure distributions for single-point DOS and TCOS optima with  $n_p = 160$  design parameters

The outcome changes markedly with multi-point optimizations. Here the number of design parameters is fixed at  $n_p = 80$ , and two multi-point optimizations are performed. The 3-point optimization adds two points with Mach numbers 0.82 and 0.86 ( $\pm 0.02$  to design point). The 6-point optimization adds additional two points

at lift coefficients 0.15 and 0.25 ( $\pm 0.05$  to design point), and one more point at Mach 0.82 and lift coefficient of 0.25 (a “diagonal” point). The convergence curves, together with the single-point optimization, are shown on figure 6. It can be seen that DOS not only starts to lose out to TCOS in run time as number of flight points is increased, but that it also heads towards increasingly worse optima. In this series of runs, the number of trimming constraints posed to the optimizer in DOS increases, from 2 in single-point, over 6 in 3-point, to 12 in 6-point optimization, and that apparently significantly complicates the problem. TCOS, on the other hand, appears not to experience any difficulty from the increasing number of flight points.



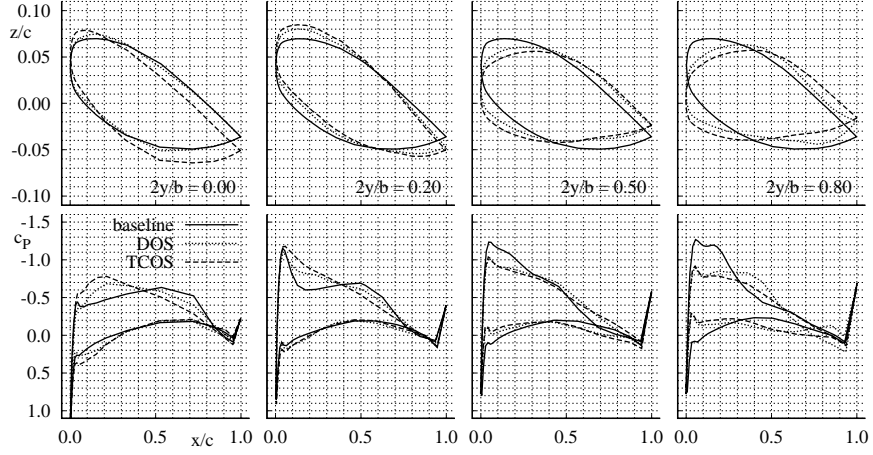
**Fig. 5** Comparison of DOS and TCOS multi-point optimization convergence with varying number of flight points  $n_{fp}$  and  $n_p = 80$  design parameters. Dots on DOS curve represent feasible designs.

Figures 6 and 7 compares the airfoil sections coming out of the multi-point optimizations. DOS and TCOS shapes now differ more than when varying number of design parameters, and increasingly so with number of flight points. DOS shapes exhibit higher variation, as well as more irregular pressure distributions.

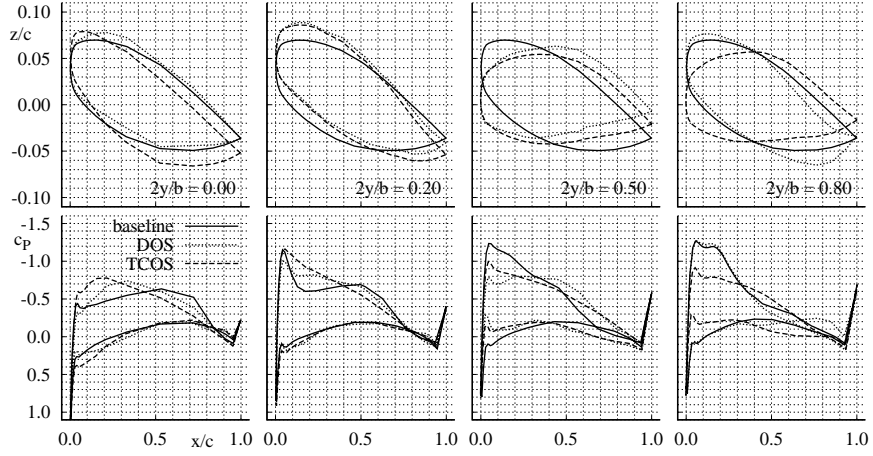
## 4 Conclusion

A comparison of DOS and TCOS optimizations has been performed, using a test case of a standalone Onera M6 wing in transonic flow. The objective was to minimize drag, and trimming was performed for the lift and pitching moment balance equations. Two series of studies were performed: one with variable number of design parameters, and another with variable number of flight points. For single-point optimizations, DOS and TCOS reached approximately the same quality of designs, with TCOS taking about 50% more run time, independent of the number of design parameters. For multi-point optimizations, as the number of flight points increased DOS was both slowing down and heading towards worse optima, while TCOS took no negative effect. Based on this, it is recommended to use TCOS on problems that have not been already well-researched and known to work well with DOS.





**Fig. 6** Comparison of airfoils and pressure distributions for 3-point multi-point DOS and TCOS optima with  $n_p = 80$  design parameters



**Fig. 7** Comparison of airfoils and pressure distributions for 6-point multi-point DOS and TCOS optima with  $n_p = 80$  design parameters

TCOS may especially be favorable when optimizations are performed “in production”, for producing actual designs, rather than as research into optimization methodology. Here also other elements counter-productive to DOS may come to front, such as reduced accuracy of gradients, or needing feasible designs quickly.

The specificity of the test case used in this report was that optimal designs were far away from the baseline point (baseline  $C_D = 0.0221$  vs approximately 0.0170 when optimized), and that there was high coupling between the shape change and control surface deflection (wing trailing edge). It is conjectured here that a conventional wing-body-tail configuration presents an easier problem from the point of

view of satisfying trimming constraints, which is supported by no problems with DOS having been seen in some studies (e.g. [2]). Therefore the next step is to perform a similar study on a long-range transport aircraft of good baseline design, such as the Common Research Model (CRM) [11].

Other possible directions are performing a similar series of tests with intentionally inaccurate gradient information, as well as with active engines.

## References

1. Brezillon, J., Brodersen, O., Dwight, R.P., Ronzheimer, A., Wild, J.: Development and application of a flexible and efficient environment for aerodynamic shape optimisation. In: ONERA-DLR Aerospace Symposium (ODAS) (2006)
2. Chen, S., Lyu, Z., Kenway, G.K.W., Martins, J.R.R.A.: Aerodynamic shape optimization of the Common Research Model wing-body-tail configuration. *Journal of Aircraft* (2015). DOI 10.2514/1.C033328
3. Dwight, R.: Robust mesh deformation using the linear elasticity equations. In: *Computational Fluid Dynamics 2006*, pp. 401–406. Springer (2006)
4. Dwight, R.P., Brezillon, J.: Efficient and robust algorithms for solution of the adjoint compressible navier-stokes equations with applications. *International Journal of Numerical Methods in Fluids* **60**(4), 365–389 (2008)
5. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization* **12**, 979–1006 (1997)
6. Merle, A., Stück, A., Rempke, A.: Discrete-adjoint based aerodynamic shape optimization process for a trimmed aircraft with active engines. In: *20th STAB/DGLR Symposium* (2016)
7. Perez, R.E., Jansen, P.W., Martins, J.R.R.A.: pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization. *Structures and Multidisciplinary Optimization* **45**(1), 101–118 (2012). DOI 10.1007/s00158-011-0666-3
8. Ronzheimer, A.: Shape based on freeform deformation in aerodynamic design optimization. In: *ERCOFTAC Design Optimization International Conference* (2004)
9. Schmitt, V., Charpin, F.: Pressure distributions on the ONERA-M6-wing at transonic Mach numbers. Tech. rep., Office National d’Etudes et Recherches Aerospatiales (ONERA) (1979)
10. Schwaborn, D., Gerhold, T., Heinrich, R.: The DLR TAU-Code: Recent Applications in Research and Industry. In: *ECCOMAS CFD 2006* (2006)
11. Vassberg, J.C., DeHaan, M.A., Rivers, S.M., Wahls, R.A.: Development of a Common Research Model for applied CFD validation studies. In: *26th AIAA Applied Aerodynamics Conference*, AIAA-2008-6919 (2008)
12. Widhalm, M., Brezillon, J., Ilić, Č., Leicht, T.: Investigation on Adjoint Based Gradient Computations for Realistic 3D Aero-Optimization. In: *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference* (2010)